AMENDMENT TO THE SPECIFICATION

Please replace the paragraph beginning on page 2, line 3, with the following amended paragraph:

Generally, the scaling of an image matrix M_1 x N_1 is scaled to a smaller size M_2 x N_2 as follows. The scaling ratios M_2/M_1 and N_2/N_1 determine the procedure in the calculation operation. If scaling takes place in real time, i.e. As a continuous flow, memory need not be reserved for the input matrix, only three memory lines being sufficient. Consider the data coming in the X lines. The first memory line sums the amount according to the scaling ~~ration~~ ratio in the X direction at the same time as the value of each pixel is ~~summer~~ summed in the Y line memory. If the scaling ratio results in the number of pixels not being an integer, the value of the pixel at the limit is weighted and summed with the two adjacent input pixels. In the same way, pixel values are calculated according to the scaling ~~ration~~ ratio into the Y line memory and, in the case of boundary pixels, they are divided weighted into two parts. When the counter set for the Y scaling ~~ration~~ ratio shows that the Y line memory is full, it is emptied forwards, after which summing starts from the beginning.

Please replace the paragraph beginning on page 4, line 9, with the following amended paragraph:

The method according to the invention includes two scaling stages, Figure 1. The first, coarse stage is simple and may comprise only the ratio 1/X. The next stage (fine) is more flexible, and may comprise the ratios Y/Z, in which $Y < Z$. X, Y, and Z are integers. The total scaling ~~ration~~ ratio is the result of the scaling ratio in both stages. The smaller the first scaling ratio (Note: $1/3 < 1/2$), the less memory will be required in the second stage. A smaller scaling ~~ration~~ ratio in the first stage will also reduce the computational logistics and the total number of calculations. The first stage can be shown in an analog or digital form. The second stage defines the memory requirement. If the scaling ~~ration~~ ratio is not directly 1/X, a better image quality will be achieved by using a smaller ratio in the second stage, but this will demand a larger memory.

Please replace the paragraph beginning on page 5, line 5, with the following amended paragraph:

In Figure 2b, the same reference numbers as in Figure 2a are used for components that are functionally similar. In the solution of the figure, the actual ~~camara~~ camera module is slightly simpler, as the fine scaling 18 has been moved to the ~~most~~ host system 22. The scaler 16' of the camera module includes only a coarse scaler 17. In this case the memory requirement is half a line in the coarse scaler (in the camera module) and three lines in the fine scaler (in the host module). For a sensor 1152x864, one line of memory represents Cx1152 words, in which C is the number of colour components (generally 3 B for RGB or YUV images). The length of the word depends on the accuracy of the calculation and is, for example, 2 or 4 bytes.

Please replace the paragraph beginning on page 7, line 25, with the following amended paragraph:

AVESKIP:

IR = MAX(Hin / Hout, Vin / Vout), in which horizontal (H) and vertical (V) sizes are used.

AVESKIP = Floor(IR)

PIXELSTEP:

MAXSTEP = 256 (or 65536 if more precise pixel positioning is desired)

PIXELSTEP = Floor((MAXSTEP * AVESKIP)/IR)

Calculation example, scaling ratio (SCRatio) ~~0,182~~ 0.182 i.e. ITR = ~~5,5~~ 5.5:

AVESKIP = Floor(~~5,5~~ 5.5) = 5

MAXSTEP = 256

PIXELSTEP = Floor (256 * 5 /~~5,5~~ 5.5) = 232

Please replace the paragraph beginning on page 8, line 1, with the following amended paragraph:

In the case of Figure 4b, the first stage scales the image as much as possible, using the ratio 1/X, in which X is the power of two (2, 4, 8, 16, 32, 64 etc.). The second stage carries out fine scaling, using as little memory as possible. This means that the scaling ~~ration~~ ratio is between [1/2, 1] and thus three lines of memory are required.

Please replace the paragraph beginning on page 8, line 7, with the following amended paragraph:

In the following table, the stages of the scaling of Figure 4b are shown are numerical values. Scaling of this kind is used in, for example, zooming, in which the resolution of the initial image is 128 x 96. In the size of the part image is greater, it is scaled to this size. In the table, the original 1-megapixel image 1152 x 864 is scaled using the ratio ~~0,111~~ 0.111 (1/8 x 128/144, index 64). In Figure 4b, the index value on the X-axis runs in the range 1 B 64 and the scaling ratio between ~~1,0~~ 1.0 - ~~0,111~~ 0.111.

Please replace the paragraph beginning on page 10, line 1, with the following amended paragraph:

AVESKIP = 2^SKIP

PIXELSTEP:

MAXSTEP = 256 (or 65536 if more precise pixel positioning is desired)

PIXELSTEP = Floor((MAXSTEP * AVESKIP) / IR)

Calculation example, ITR = ~~5,5~~ 5.5:

SKIP = Floor (LOG2(ITR)) = Floor (~~2,46~~ 2.46) = 2

AVESKIP = 2^2 = 4

PIXELSTEP = Floor (256 * 4 /~~5,5~~ 5.5) = 186

Please replace the paragraph beginning on page 10, line 15, with the following amended paragraph:

The integer calculation auxiliary variables AVESKIP and PIXELSTEP are defined in the following:

AVESKIP:

Inverted total scaling ratio IR = MAX(Hin / Hout, Vin / Vout), in which horizontal (H) and vertical (V) sizes are used.

AVESKIP = Floor(Sqrt(IR))

PIXELSTEP:

MAXSTEP = 256 (or 65536, if more precise pixel positioning is desired)

PIXELSTEP = Floor(MAXSTEP * AVESKIP) / IR)

Calculation example 3, ITR = ~~5,5~~ 5.5:

AVESKIP = Floor (Sqrt (~~5,5~~ 5.5)) = 2

MAXSTEP = 256

PIXELSTEP = Floor (256 * 2 /~~5,5~~ 5.5) = 93

Please replace the Table beginning on page 9, line 1, with the following amended Table:

X-size	Y-size	ratio X	X	Y	Z	index
128	96	~~1,000~~ 1.000	1	128	128	1
132	99	~~0,970~~ 0.970	1	128	132	2
137	103	~~0,934~~ 0.934	1	128	137	3
141	106	~~0,908~~ 0.908	1	128	141	4
146	110	~~0,877~~ 0.877	1	128	146	5
151	114	~~0,848~~ 0.848	1	128	151	6
157	118	~~0,815~~ 0.815	1	128	157	7

162	122	~~0,790~~ 0.790	1	128	162	8
168	126	~~0,762~~ 0.762	1	128	168	9
174	131	~~0,736~~ 0.736	1	128	174	10
180	135	~~0,711~~ 0.711	1	128	180	11
187	140	~~0,684~~ 0.684	1	128	187	12
193	145	~~0,663~~ 0.663	1	128	193	13
200	150	~~0,640~~ 0.640	1	128	200	14
208	156	~~0,615~~ 0.615	1	128	208	15
215	161	~~0,595~~ 0.595	1	128	215	16
222	167	~~0,577~~ 0.577	1	128	222	17
230	173	~~0,557~~ 0.557	1	128	230	18
237	177	~~0,540~~ 0.540	1	128	237	19
246	184	~~0,520~~ 0.520	1	128	246	20
256	192	~~0,500~~ 0.500	2	128	128	21
264	198	~~0,485~~ 0.485	2	128	132	22
274	206	~~0,467~~ 0.467	2	128	137	23
282	212	~~0,454~~ 0.454	2	128	141	24
292	220	~~0,438~~ 0.438	2	128	146	25
302	228	~~0,424~~ 0.424	2	128	151	26
314	236	~~0,408~~ 0.408	2	128	157	27
324	244	~~0,395~~ 0.395	2	128	162	28
336	252	~~0,381~~ 0.381	2	128	168	29
348	262	~~0,368~~ 0.368	2	128	174	30
360	270	~~0,356~~ 0.356	2	128	180	31
374	280	~~0,342~~ 0.342	2	128	187	32
386	290	~~0,332~~ 0.332	2	128	193	33
400	300	~~0,320~~ 0.320	2	128	200	34
416	312	~~0,308~~ 0.308	2	128	208	35
430	322	~~0,298~~ 0.298	2	128	215	36
444	334	~~0,288~~ 0.288	2	128	222	37
460	346	~~0,278~~ 0.278	2	128	230	38

474	354	~~0,270~~ 0.270	2	128	237	39
492	368	~~0,260~~ 0.260	2	128	246	40
512	384	~~0,250~~ 0.250	4	128	128	41
528	396	~~0,242~~ 0.242	4	128	132	42
548	412	~~0,234~~ 0.234	4	128	137	43
564	424	~~0,227~~ 0.227	4	128	141	44
584	440	~~0,219~~ 0.219	4	128	146	45
604	456	~~0,212~~ 0.212	4	128	151	46
628	472	~~0,204~~ 0.204	4	128	157	47
648	488	~~0,198~~ 0.198	4	128	162	48
672	504	~~0,190~~ 0.190	4	128	168	49
696	524	~~0,184~~ 0.184	4	128	174	50
720	540	~~0,178~~ 0.178	4	128	180	51
748	560	~~0,171~~ 0.171	4	128	187	52
772	580	~~0,166~~ 0.166	4	128	193	53
800	600	~~0,160~~ 0.160	4	128	200	54
832	624	~~0,154~~ 0.154	4	128	208	55
860	644	~~0,149~~ 0.149	4	128	215	56
888	668	~~0,144~~ 0.144	4	128	222	57
920	692	~~0,139~~ 0.139	4	128	230	58
948	708	~~0,135~~ 0.135	4	128	237	59
984	736	~~0,130~~ 0.130	4	128	246	60
1024	768	~~0,125~~ 0.125	8	128	128	61
1056	792	~~0,121~~ 0.121	8	128	132	62
1104	832	~~0,116~~ 0.116	8	128	138	63
1152	864	~~0,111~~ 0.111	8	128	144	64